

# PATENT APPLICATION

## REDUCING CPU OVERHEAD IN THE FORWARDING PROCESS

Inventors: Hon Wah Chin  
3281 Greer Road  
Palo Alto, California 94303  
Citizenship: United States

Assignee: Cisco Technology, Inc.  
170 West Tasman Drive  
San Jose, California 95134-1706

A corporation of California

Prepared by:

BEYER & WEAVER, LLP  
P.O. Box 61059  
Palo Alto, CA 94306  
Telephone (650) 493-2100

**REDUCING CPU OVERHEAD IN THE  
FORWARDING PROCESS**

5

**BACKGROUND OF THE INVENTION**

**1. Field of the Invention**

The present invention relates to packet forwarding. More particularly, the present invention relates to forwarding packets through manipulating multiple queues of packets.

**2. Description of the Related Art**

Routers are commonly used to forward packets within a network. Routers typically include a CPU, a memory, and a plurality of inbound and outbound ports. Typically, data is received at an inbound port as a packet. When a packet is received at an inbound port, it is ultimately filtered or forwarded via an outbound port.

FIG. 1 is a block diagram illustrating a conventional router 102. Rather than sending each packet as it is received at an inbound port 104-1, 104-2, packets are typically stored in a single inbound queue 106-1, 106-2 via a hardware inbound interface, or "controller," associated with the corresponding inbound port 104-1, 104-2. When the CPU wakes up, it obtains a packet from the inbound queue 104-1, 104-2 classifies, and transfers the packet to a hardware outbound interface, or "controller." The hardware outbound interface then stores the packet in an outbound queue 108-1, 108-2 associated with an outbound port 110-1, 110-2. Transferring may be accomplished through

moving the packet or simply modifying a pointer to the packet. The outbound interface then forwards the packet. Typically, this is performed for each packet.

In order to send each packet stored in the inbound queue, the packets must be transferred to the appropriate outbound queue. Often, each peripheral interface is designed for use with a particular protocol (e.g., Ethernet) and therefore a particular representation, or data structure. Thus, the inbound interface and the outbound interface often have incompatible data structures. By way of example, queue descriptions for the inbound interface and the outbound interface often have different formats. Thus, the CPU must translate the packets stored in the inbound queue to an outbound representation compatible with the outbound interface. The packet is then transferred to an outbound queue associated with an outbound port. Accordingly, it would be desirable if the inbound interface and the outbound interface could be designed to provide compatible data structures, thereby facilitating the transfer of data between the inbound and the outbound controller.

Inbound controllers typically store packets as they are received in a single inbound queue. FIG. 2 is a block diagram illustrating a router having a conventional inbound and outbound controller. As shown, data 202 is received by an inbound controller 204. The inbound controller 204 then stores this data 202 (e.g., packet) in memory 206 in an inbound queue 208. The inbound queue 208 includes a plurality of queue entries 210 corresponding to a plurality of packets. By way of example, each one of the plurality of packets may be stored in a packet buffer 212. In order to forward each packet stored in the inbound queue 208, the CPU must transfer each entry 210 in the inbound queue 208 to an outbound queue 214 associated with an outbound controller 216. The outbound queue 214 similarly includes a plurality of queue entries 218 corresponding to a plurality of packets. Thus, an entry 210 in the inbound queue 208 may be transferred to the outbound queue 214 through modifying a pointer to a packet buffer 220 associated with an entry being transferred.

Since each packet is individually transferred from the inbound controller to the outbound controller, this produces a substantial burden on the CPU and a low packet per second forwarding rate. By way of example, each queue entry in both the inbound and the outbound queues may include a packet descriptor and a pointer to a packet buffer that holds data for the packet. Thus, transferring a single packet may require that a pointer to the transferred packet be added to the outbound queue as well as removed from the inbound queue. In addition, any packet descriptors must similarly be transferred. Since the transfer of each packet requires that several steps be performed, the burden on the CPU is substantial. Thus, it would be desirable if an inbound interface for a router could be designed which would reduce this burden on the CPU while maximizing the packet per second forwarding rate. Accordingly, it would be beneficial if entire queues could be transferred between the inbound controller and the outbound controller rather than individual packets, thereby increasing the throughput of a router.

Classification of packets similarly increases the burden on the CPU. While inbound controllers and outbound controllers are typically implemented in hardware, classification of packets is typically performed in software. Thus, numerous CPU cycles are commonly dedicated to such classification. Accordingly, it would be beneficial if an inbound interface could be implemented in hardware such that each inbound packet were sorted into an inbound queue corresponding sorting criteria such as the source address and the destination address of the packet.

Once each packet is sent by the outbound controller, the CPU often deallocates the corresponding packet buffer. Thus, this deallocation is typically performed for each packet individually. However, it would be beneficial if the memory could be deallocated and reused in a more efficient manner. Accordingly, it would be desirable if an outbound interface for a router could be designed which would reduce the number of CPU cycles dedicated to the deallocation of these buffers.

In addition to dedicating numerous CPU cycles to transferring packets between an inbound controller and an outbound controller, classifying packets, and deallocating packet buffers associated with the packets, these packet buffers typically consume a considerable amount of memory. A packet buffer is typically preallocated for each entry in an inbound queue. Since all of these packet buffers may not be utilized consistently, a portion of these dedicated buffers often remain unused. Accordingly, it would be desirable to reduce the amount of memory required and therefore the production costs associated with each router.

## SUMMARY OF THE INVENTION

The present invention provides methods and apparatus for transferring a queue of packets from an inbound port to an outbound port within a router. This is accomplished through providing an inbound controller and an outbound controller which are controlled by a CPU. In this manner, a plurality of packets may be simultaneously transferred between the inbound controller and the outbound controller, thereby reducing CPU overhead in the forwarding process.

According to one aspect of the invention, a router includes an inbound controller. The router includes a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU. The inbound controller is adapted for receiving an inbound packet at one of the plurality of inbound ports. A plurality of inbound queues are provided for one of the plurality of inbound ports. An inbound packet is then received at the one of the plurality of inbound ports. A classifier classifies the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria. The inbound packet is then stored in the selected one of the plurality of inbound queues. The inbound controller is capable of determining when one of the plurality of inbound queues is ready to be moved to an outbound queue which is capable of storing a multiplicity of inbound queues.

According to another aspect of the invention, an outbound controller is provided. The outbound controller is adapted for forwarding packets at one of the plurality of outbound ports of a router. At least one outbound queue is provided for one of the plurality of outbound ports which is capable of storing a plurality of inbound queues. The outbound controller is capable of receiving a notification to handle an inbound queue storing a plurality of packets. The inbound queue is then transferred to an outbound queue. By way of example, the outbound controller may include an outbound classifier that classifies the inbound queue in one of a plurality of outbound queues.

According to yet another aspect of the invention, the present invention may be implemented in an encryption system. The encryption system includes an inbound controller adapted for receiving an inbound packet. A first classifier is coupled to the inbound controller and adapted for classifying and storing the inbound packet in an inbound queue. An outbound controller is adapted for receiving the inbound queue. In addition, an encryption box is coupled to the outbound controller. The encryption box is adapted for encrypting the inbound queue to provide an encrypted inbound queue to the outbound controller for transmission. In addition, the outbound controller may include a second classifier that classifies the encrypted inbound queue in an outbound queue. The outbound controller may then transmit data stored in the outbound queue.

A CPU controls the inbound controller and the outbound controller. Thus, through combining software and hardware modules, the present invention provides efficient and accurate packet forwarding. Hardware may be utilized to provide speed where global knowledge is not required. By way of example, a hardware classifier may provide an increased forwarding rate. At the same time, software provides global knowledge and intelligence in forwarding decisions to increase the efficiency of the forwarding process.

**BRIEF DESCRIPTION OF THE DRAWINGS**

FIG. 1 is a block diagram illustrating a conventional router.

5 FIG. 2 is a block diagram illustrating a router having a conventional inbound and outbound controller.

FIG. 3 is a block diagram illustrating a router according to one embodiment of the invention.

10 FIG. 4 is a block diagram illustrating an exemplary data structure that may be used for each inbound queue according to one embodiment of the invention.

FIG. 5 is a block diagram illustrating a system including an inbound controller and an outbound controller according to one embodiment of the invention.

15 FIG. 6 is a flow diagram illustrating a method for implementing an inbound controller according to one embodiment of the present invention.

FIG. 7 is a flow diagram illustrating the method for classifying inbound packets of FIG. 6 according to one embodiment of the present invention.

20 FIG. 8 is a diagram illustrating an exemplary address table that may be used to classify each packet according to one embodiment of the invention.

FIG. 9 is a flow diagram illustrating the method of storing an inbound packet in an inbound queue of FIG. 6 according to one embodiment of the invention.

25 FIG. 10 is a flow diagram illustrating the method of determining whether one of the plurality of inbound queues is ready to be moved to an outbound queue, shown in FIG. 6, according to one embodiment of the invention.

FIG. 11 is a flow diagram illustrating one method for operation of the CPU in response to the inbound controller according to one embodiment of the invention.

5 FIG. 12 is a flow diagram illustrating a method for implementing an outbound controller according to one embodiment of the invention

FIG. 13 is a flow diagram illustrating the method for transferring the inbound queue to an outbound queue, shown in FIG. 12, according to one embodiment of the invention.

FIG. 14 is a flow diagram illustrating the method for transmitting packets of FIG. 12 according to one embodiment of the invention.

FIG. 15 is a flow diagram illustrating one method for operation of the CPU in response to the outbound controller according to one embodiment of the invention.

FIG. 16 is a block diagram illustrating a conventional encryption system.

FIG. 17 is a block diagram illustrating an encryption system according to one embodiment of the invention.

## DETAILED DESCRIPTION OF THE INVENTION

5 In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be obvious, however, to one skilled in the art, that the present invention may be practiced without some or all of these specific details. In other instances, well known process steps have not been described in detail in order not to unnecessarily obscure the present invention.

An invention is described herein that reduces the burden on the CPU and therefore increases the packet per second forwarding rate within a router. This is accomplished through providing an inbound controller and an outbound controller which store packets in compatible formats such that a set of packets (e.g., queue of packets) may be transferred between the inbound controller and the outbound controller. In accordance with one embodiment, the present invention provides multiple inbound queues associated with each inbound port. Each of these inbound queues corresponds to a set of packet sorting criteria (e.g., destination address and/or priority). The inbound interface then sorts inbound packets received at an inbound port into one of these inbound queues according to packet sorting data associated with the packet sorting criteria. By way of example, the destination address of the packet as well as the priority of the packet may be obtained and utilized to select the appropriate inbound queue. An outbound queue is provided for each outbound port. The CPU may then transfer each inbound queue to the outbound queue corresponding to the inbound queue classification (e.g., destination address of the packet) such the inbound queues are sorted according to the priority of the inbound queue.

20 Referring now to FIG. 3, a block diagram of a router according to one embodiment of the invention is illustrated. As shown, a plurality of inbound queues 302, 304 are provided for each inbound port 306 and an outbound

queue 308 is provided for each outbound port 310. The inbound 302, 304 and 5  
outbound 308 queues are stored in memory 312 such that the inbound queues  
302, 304 are accessible to an inbound controller 314 and the outbound queues  
308 are accessible to an outbound controller 316. The inbound queues 302,  
304 are created such that they permit classification of inbound packets  
corresponding to packet sorting criteria. By way of example, the packet sorting  
criteria may include the source address of the inbound packet, the destination  
address of the inbound packet, the priority of the packet, and/or other criteria.  
Each inbound queue includes a plurality of queue entries 318, each of which  
10 correspond to a single packet. By way of example, each queue entry may  
include a packet descriptor and a pointer to a packet buffer that holds data for  
the packet. The packet descriptors may be pre-allocated while the packet  
buffers may be obtained from a free pool of available memory. In order to  
forward each packet, the CPU transfers each inbound queue 302, 304 to the  
outbound controller 316. The outbound controller 316 then stores the inbound  
queue in an entry 320 in the appropriate outbound queue 308. By way of  
example, the outbound controller 316 may have an internal pointer to each  
15 outbound queue 308. Thus, the outbound queue 308 includes a plurality of  
queue entries 320 corresponding to a set of inbound queues. By way of  
example, the outbound queue may include a first-in-first-out (FIFO) of inbound  
queue headers with pointers to the transferred inbound queues. Although one  
20 outbound queue 308 is shown for each outbound port 310, multiple outbound  
queues may be provided.

Various data structures may be utilized in the implementation of the  
25 inbound and the outbound queues. One exemplary data structure that may be  
used for each inbound queue according to one embodiment of the invention is  
illustrated in FIG. 4. As shown, each queue 402 may contain a queue header  
404 that includes a number of packets 406 in the queue and a queue length 408  
(e.g., number of aggregate bytes in the queue). The queue header 404 may  
30 further include a maximum number of queue entries that may be stored in the  
queue, or amount of memory allocated for packet headers (not shown to

5 simplify illustration). The information stored in the queue header 404 may be used by the inbound controller to determine when each inbound queue will be transferred to the outbound controller. By way of example, when an inbound queue reaches a specified number of packets or contains a specified number of bytes, the inbound queue may be transferred from the inbound controller to the 10 outbound controller. Thus, this queue header 404 may be ignored by the outbound controller. Alternatively, the queue header 404 for an inbound queue may further include a queue ID corresponding to the appropriate outbound queue (not shown to simplify illustration). Each entry in the queue 15 402 may include a packet header 410, or descriptor, which includes a packet length 412 (e.g., number of bytes in the packet) as well as a status bit 414. A similar status bit may be implemented in the queue header 404 to indicate the status of an entire queue (not shown to simplify illustration). By way of example, the status bit 414 may indicate that a packet, or queue, is valid, invalid, or sent. Each queue entry may further include a pointer to packet data 20 416. The inbound and outbound queues may be implemented as linked lists, arrays or other data structures that may be utilized to store a set of packets.

25 Each inbound queue may be transferred to the outbound controller through the assertion of an interrupt. Once transferred, the inbound queue will be stored in the appropriate outbound queue by an outbound controller. Thus, each outbound queue may contain a set of pointers to the inbound queues received by the outbound controller.

30 Various configurations in which the inbound and the outbound controller may be implemented are possible. By way of example, one inbound controller and one outbound controller may be provided for the entire router.

35 Alternatively, as shown in FIG. 5, the present invention may include one inbound controller for each inbound port and one outbound controller for each outbound port. A packet stream may be received by a media access controller (MAC) 502 where the present invention is implemented in a broadcast network. In addition, an inbound controller 504 is shown to include a classifier 506 to

5        permit classification of packets in an appropriate inbound queue. However, the classifier 506 may be implemented separately from the inbound controller 504. A memory 508 is provided for use by the inbound controller 504 and an  
10      outbound controller 512. In addition, a CPU 510 is provided which supervises the inbound controller 504 and the outbound controller 512. The memory 508 may store compatible representations of the inbound queues and the outbound queues, as well as pools of available memory, for use by the inbound controller 504 and the outbound controller 512. Thus, the outbound controller 512 may store each inbound queue received from the inbound controller in an entry in the appropriate outbound queue. The outbound controller 512 may then transmit packets stored in the outbound queue. Thus, a packet stream is provided to the corresponding outbound port.

15      One method for implementing an inbound controller according to one embodiment of the present invention is illustrated in FIG. 6. As described above, a plurality of inbound queues are provided for one of the plurality of inbound ports. At step 602, an inbound packet is received at the one of the plurality of inbound ports. The inbound packet is then classified in a selected one of the plurality of inbound queues according to packet sorting criteria at step 604. By way of example, the inbound packet (e.g., packet header) may include a source address and a destination address. Thus, the packet sorting criteria may include the source address and the destination address of the inbound packet, as well as other sorting criteria. The inbound packet is then stored in the selected one of the plurality of inbound queues at step 606. In addition, the queue header may be updated to reflect the current number of  
20      packets and bytes stored within the inbound queue. Next, it is determined whether one of the plurality of inbound queues is ready to be moved to an outbound queue at step 608. By way of example, it is determined whether to wake up the CPU. Rather than wake up the CPU upon receiving each packet, the CPU is woken up at various intervals. If it is determined to wake up the  
25      CPU at step 610, an interrupt is asserted at step 612 to signal when one of the plurality of inbound queues is ready to be moved by the CPU to an outbound  
30      queue.

queue. If it is not determined to wake up the CPU at step 610, the inbound controller receives the next inbound packet at step 602. Although the inbound controller is described as being implemented for a single port, the inbound controller may be implemented for multiple ports or an entire router.

5           In order to transfer each inbound queue to the appropriate outbound queue, each packet must be classified in the appropriate inbound queue. FIG. 7 is a flow diagram illustrating the method for classifying inbound packets 604 of FIG. 6 according to one embodiment of the present invention. Each inbound packet may be sorted according to various sorting criteria. At step 702, inbound packet sorting criteria are selected to permit each inbound packet to be classified in one of the inbound queues. The packet sorting criteria may include the source address of the inbound packet, the destination address of the inbound packet, the priority of the packet (e.g., session ID), the IP address of the packet, and/or other criteria such as information which may be obtained from the packet. Next, at step 704, packet sorting data associated with the packet sorting criteria is obtained for the inbound packet being sorted. By way of example, the destination address of the inbound packet may be obtained. Next, at step 706, the inbound packet is sorted into one of the inbound queues according to the packet sorting data. By way of example, an address table including each inbound queue may be provided that includes the packet sorting data associated with each queue. Thus, the inbound packet may be classified upon locating the relevant packet sorting data in the address table. By way of example, a classified inbound packet may be associated with an inbound queue through storing a queue ID of the inbound queue in the inbound packet (e.g., packet header). While classification of packets may be performed for various purposes, such classification is typically performed in software. Accordingly, in one embodiment of the invention, the classifier is implemented in hardware to maximize the packet per second forwarding rate.

10

15

20

25

30

As described above, one method for classification of inbound packets is through the use of one or more address tables. FIG. 8 is a diagram illustrating

an exemplary address table that may be used to classify each packet according to one embodiment of the invention. A separate classifier and address table may be provided for each port. In addition, software may be provided which allows the CPU to create the address tables required by each classifier.

5 An exemplary address table 802 may include a destination address 804 or outbound port number of the packet, a priority 806 of the packet, and a queue ID 808 corresponding to an inbound queue. The inbound queue classification will further correspond to an outbound queue. By way of example, the queue ID of the corresponding outbound queue may be stored in the queue header of each inbound queue. Thus, the outbound queue may be implicit in the inbound queue in which each packet is classified. As described above, packet classification may be accomplished through the use of a different address table for each port, as shown. Alternatively, a single global address table may be used.

10 15 In order to optimize the use of available memory, a free pool of available memory buffers may be maintained rather than pre-allocating memory. By way of example, this free pool may be implemented through the use of a linked list, array, or other data structure compatible with the inbound queues and the outbound queues. Moreover, multiple free pools may be utilized to recycle 20 available packet buffers, packet descriptors, or queue descriptors.

Prior to storing each classified packet in the appropriate inbound queue, memory may be allocated from one or more free pools of available memory. FIG. 9 is a flow diagram illustrating the method of storing an inbound packet in an inbound queue 606 of FIG. 6 according to one embodiment of the invention. At step 902, an available packet buffer may be obtained from a free pool of 25 available packet buffers. Thus, memory may be obtained to permit storage of the inbound packet. The inbound packet is then stored in the available packet buffer at step 904. The packet buffer containing the inbound packet is then stored in the inbound queue at step 906. By way of example, a pointer to the

packet buffer containing the inbound packet may be stored in the inbound queue.

Rather than wake up the CPU upon receiving each packet, the CPU is woken up at various intervals. FIG. 10 illustrates the method of determining whether to wake up the CPU 608 of FIG. 6 according to one embodiment of the invention. At step 1002, it is determined whether the number of packets in any one of the inbound queues exceeds a maximum number of packets. By way of example, a maximum number of packets may be specified for each inbound queue. As yet another example, a single global maximum number of packets may be specified for the inbound queues. If the maximum number of packets is exceeded, it has been determined to wake up the CPU at step 1004. However, if the maximum number of packets has not been exceeded, it is next determined whether the length of any one of the inbound queues (e.g., number of aggregate bytes within the queue) exceeds a maximum queue length at step 1006. If the maximum queue length is exceeded, it has been determined to wake up the CPU at step 1008. However, if the maximum queue length has not been exceeded, it is next determined whether the free pool of memory has been depleted at step 1010. By way of example, a desired minimum amount of available memory may be specified to indicate depletion of one or more free pools of memory. If the free pool has been depleted, it has been determined to wake up the CPU at step 1012. However, if the free pool has not been depleted, it is next determined whether a maximum time limit (e.g., 0.5 milliseconds) has been exceeded at step 1014. In instances where network traffic is slow, inbound queues will fill slowly and a large portion of the free pool may remain available. Therefore, the maximum time limit sets a minimum forwarding rate in instances where inbound traffic is at a minimum. The above determinations may be made separately or in combination to determine whether to wake up the CPU through the assertion of an interrupt. Moreover, these determinations are exemplary, and therefore further determinations may be made.

Upon the assertion of an interrupt, the CPU may transfer an inbound queue to an outbound controller or outbound queue associated with an outbound port. FIG. 11 illustrates one method for operation of the CPU in response to the inbound controller according to one embodiment of the invention. At step 1102, an inbound queue may be obtained from the inbound controller. The obtained inbound queue may then transferred to the outbound controller at step 1104. A new inbound queue may then be provided to the inbound controller as a replacement for the obtained inbound queue at step 1106. This new inbound queue may be obtained from the outbound controller, as will be further described below, or may be obtained from a free pool of available memory. Moreover, the CPU may not need to act to provide this replacement queue when inbound and outbound traffic are balanced, since a new queue will automatically be made available as packets are forwarded through each port. In this manner, the CPU transfers a queue of packets from the inbound controller to the outbound controller.

The outbound controller stores each received inbound queue in an outbound queue and selectively transmits packets stored in the outbound queue. FIG. 12 is a flow diagram illustrating one method for implementing an outbound controller according to one embodiment of the invention. A notification may be received from the CPU to handle an inbound queue at step 1202. As described above, the outbound controller may be implemented for one of the plurality of outbound ports. Moreover, an outbound queue associated with the one of the plurality of outbound ports is provided. The outbound queue is configured such that it is capable of storing a plurality of inbound queues. By way of example, the outbound queue may include a plurality of entries capable of storing pointers to each one of the plurality of inbound queues. The inbound queue is then transferred to the outbound queue at step 1204. Packets stored in each inbound queue of the outbound queue may then be transmitted at step 1206. Although the outbound controller is described as providing one outbound queue for a single port, multiple outbound queues may be provided. Moreover,

although described as being implemented for a single port, the outbound controller may be implemented for multiple ports or an entire router.

When an inbound queue is received by the outbound controller, the outbound controller transfers the inbound queue to an outbound queue. FIG. 13 is a flow diagram illustrating the method for transferring the inbound queue to an outbound queue 1204 of FIG. 12 according to one embodiment of the invention. Since one outbound queue may be provided for each outbound port and associated outbound controller, each inbound queue may be stored in the outbound queue according to a priority order. Thus, a priority of the inbound queue may be ascertained at step 1302. By way of example, a priority of the inbound queue may be provided in the queue header for the inbound queue. As yet another example, the priority of the inbound queue may be provided in the corresponding address table. The inbound queue may then be transferred to the outbound queue according to the priority of the inbound queue at step 1304. Moreover, since multiple outbound queues may be provided, the appropriate outbound queue may be determined from an address table such as that illustrated in FIG. 8.

The outbound controller may selectively transmit packets as well as deallocate memory previously used by the transmitted packets. FIG. 14 is a flow diagram illustrating the method for transmitting packets 1206 of FIG. 12 according to one embodiment of the invention. Since the inbound queues may be stored in the outbound queue according to the priority of the inbound queues, the packets in each inbound queue may be transmitted consecutively. Prior to transmitting these packets or during transmission, data may be selectively released at step 1400. The packet buffers associated with this data may similarly be released and made available for use (e.g., returned to a free pool of buffers). By way of example, selected packets or an entire inbound queue may be tagged as sent or invalid to maximize the throughput of the router. This technique may be utilized to accommodate for a high level of traffic where the number of packets being sent is excessive. Moreover, where there is

an unacceptable delay during packet transmission (e.g., voice transmission), these packets or queues of packets may be beneficially discarded. Similarly, this technique may be beneficial where the amount of free memory available for use by the inbound and/or the outbound controller is diminishing. Thus, the 5 outbound controller may selectively transmit packets based upon the priority of the packets. By way of example, packets may be prioritized in a plurality of queues, each of the plurality of queues having a different priority.

10 Packets may be transmitted for each inbound queue in the outbound queue. Thus, a next inbound queue in the outbound queue may be obtained at step 1402. A next packet from the obtained inbound queue may then be obtained at step 1404. A status bit of the packet may then be checked at step 1406. If the status bit indicates that the packet is valid at step 1408, the status bit may be set to sent or invalid at step 1410. The packet may then be sent at step 1412. However, if the status bit of the packet indicates that the packet is invalid, a next packet may be obtained at step 1404.

15 After the packet is sent, it is determined whether there are more packets in the inbound queue at step 1414. If there are more packets, a next packet may be obtained from the inbound queue at step 1404. However, if there are no packets remaining in the inbound queue, memory (e.g., packet buffers) in the inbound queue may be released at step 1416. By way of example, the entire inbound queue may be released. Alternatively, all invalid and/or sent packet buffers may be released. The empty inbound queue may therefore be made available for use at step 1416. By way of example, the released packet buffers 20 may be stored in a free pool of available packet buffers. Moreover, the empty inbound queue may be used as a new inbound queue by the inbound controller. As yet another example, the empty inbound queue may be used by the 25 outbound controller during bi-directional (e.g., full duplex) operation. In addition, this newly available inbound queue may be stored in a separate entity having a data structure (e.g., linked list) compatible with the free pool of available memory, as well as the inbound and outbound queues. The CPU may

5 therefore reuse the memory for each inbound queue rather than recovering each available packet buffer individually. Since memory may be reused by the inbound controller and the outbound controller, the CPU may dedicate a majority of the transfer operations performed to the transmission of packets  
10 rather than the transfer of free buffers. Each inbound queue that remains in the outbound queue may then be processed at step 1418.

10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100  
105  
110  
115  
120  
125  
130  
135  
140  
145  
150  
155  
160  
165  
170  
175  
180  
185  
190  
195  
200  
205  
210  
215  
220  
225  
230  
235  
240  
245  
250  
255  
260  
265  
270  
275  
280  
285  
290  
295  
300  
305  
310  
315  
320  
325  
330  
335  
340  
345  
350  
355  
360  
365  
370  
375  
380  
385  
390  
395  
400  
405  
410  
415  
420  
425  
430  
435  
440  
445  
450  
455  
460  
465  
470  
475  
480  
485  
490  
495  
500  
505  
510  
515  
520  
525  
530  
535  
540  
545  
550  
555  
560  
565  
570  
575  
580  
585  
590  
595  
600  
605  
610  
615  
620  
625  
630  
635  
640  
645  
650  
655  
660  
665  
670  
675  
680  
685  
690  
695  
700  
705  
710  
715  
720  
725  
730  
735  
740  
745  
750  
755  
760  
765  
770  
775  
780  
785  
790  
795  
800  
805  
810  
815  
820  
825  
830  
835  
840  
845  
850  
855  
860  
865  
870  
875  
880  
885  
890  
895  
900  
905  
910  
915  
920  
925  
930  
935  
940  
945  
950  
955  
960  
965  
970  
975  
980  
985  
990  
995  
1000  
1005  
1010  
1015  
1020  
1025  
1030  
1035  
1040  
1045  
1050  
1055  
1060  
1065  
1070  
1075  
1080  
1085  
1090  
1095  
1100  
1105  
1110  
1115  
1120  
1125  
1130  
1135  
1140  
1145  
1150  
1155  
1160  
1165  
1170  
1175  
1180  
1185  
1190  
1195  
1200  
1205  
1210  
1215  
1220  
1225  
1230  
1235  
1240  
1245  
1250  
1255  
1260  
1265  
1270  
1275  
1280  
1285  
1290  
1295  
1300  
1305  
1310  
1315  
1320  
1325  
1330  
1335  
1340  
1345  
1350  
1355  
1360  
1365  
1370  
1375  
1380  
1385  
1390  
1395  
1400  
1405  
1410  
1415  
1420  
1425  
1430  
1435  
1440  
1445  
1450  
1455  
1460  
1465  
1470  
1475  
1480  
1485  
1490  
1495  
1500  
1505  
1510  
1515  
1520  
1525  
1530  
1535  
1540  
1545  
1550  
1555  
1560  
1565  
1570  
1575  
1580  
1585  
1590  
1595  
1600  
1605  
1610  
1615  
1620  
1625  
1630  
1635  
1640  
1645  
1650  
1655  
1660  
1665  
1670  
1675  
1680  
1685  
1690  
1695  
1700  
1705  
1710  
1715  
1720  
1725  
1730  
1735  
1740  
1745  
1750  
1755  
1760  
1765  
1770  
1775  
1780  
1785  
1790  
1795  
1800  
1805  
1810  
1815  
1820  
1825  
1830  
1835  
1840  
1845  
1850  
1855  
1860  
1865  
1870  
1875  
1880  
1885  
1890  
1895  
1900  
1905  
1910  
1915  
1920  
1925  
1930  
1935  
1940  
1945  
1950  
1955  
1960  
1965  
1970  
1975  
1980  
1985  
1990  
1995  
2000  
2005  
2010  
2015  
2020  
2025  
2030  
2035  
2040  
2045  
2050  
2055  
2060  
2065  
2070  
2075  
2080  
2085  
2090  
2095  
2100  
2105  
2110  
2115  
2120  
2125  
2130  
2135  
2140  
2145  
2150  
2155  
2160  
2165  
2170  
2175  
2180  
2185  
2190  
2195  
2200  
2205  
2210  
2215  
2220  
2225  
2230  
2235  
2240  
2245  
2250  
2255  
2260  
2265  
2270  
2275  
2280  
2285  
2290  
2295  
2300  
2305  
2310  
2315  
2320  
2325  
2330  
2335  
2340  
2345  
2350  
2355  
2360  
2365  
2370  
2375  
2380  
2385  
2390  
2395  
2400  
2405  
2410  
2415  
2420  
2425  
2430  
2435  
2440  
2445  
2450  
2455  
2460  
2465  
2470  
2475  
2480  
2485  
2490  
2495  
2500  
2505  
2510  
2515  
2520  
2525  
2530  
2535  
2540  
2545  
2550  
2555  
2560  
2565  
2570  
2575  
2580  
2585  
2590  
2595  
2600  
2605  
2610  
2615  
2620  
2625  
2630  
2635  
2640  
2645  
2650  
2655  
2660  
2665  
2670  
2675  
2680  
2685  
2690  
2695  
2700  
2705  
2710  
2715  
2720  
2725  
2730  
2735  
2740  
2745  
2750  
2755  
2760  
2765  
2770  
2775  
2780  
2785  
2790  
2795  
2800  
2805  
2810  
2815  
2820  
2825  
2830  
2835  
2840  
2845  
2850  
2855  
2860  
2865  
2870  
2875  
2880  
2885  
2890  
2895  
2900  
2905  
2910  
2915  
2920  
2925  
2930  
2935  
2940  
2945  
2950  
2955  
2960  
2965  
2970  
2975  
2980  
2985  
2990  
2995  
3000  
3005  
3010  
3015  
3020  
3025  
3030  
3035  
3040  
3045  
3050  
3055  
3060  
3065  
3070  
3075  
3080  
3085  
3090  
3095  
3100  
3105  
3110  
3115  
3120  
3125  
3130  
3135  
3140  
3145  
3150  
3155  
3160  
3165  
3170  
3175  
3180  
3185  
3190  
3195  
3200  
3205  
3210  
3215  
3220  
3225  
3230  
3235  
3240  
3245  
3250  
3255  
3260  
3265  
3270  
3275  
3280  
3285  
3290  
3295  
3300  
3305  
3310  
3315  
3320  
3325  
3330  
3335  
3340  
3345  
3350  
3355  
3360  
3365  
3370  
3375  
3380  
3385  
3390  
3395  
3400  
3405  
3410  
3415  
3420  
3425  
3430  
3435  
3440  
3445  
3450  
3455  
3460  
3465  
3470  
3475  
3480  
3485  
3490  
3495  
3500  
3505  
3510  
3515  
3520  
3525  
3530  
3535  
3540  
3545  
3550  
3555  
3560  
3565  
3570  
3575  
3580  
3585  
3590  
3595  
3600  
3605  
3610  
3615  
3620  
3625  
3630  
3635  
3640  
3645  
3650  
3655  
3660  
3665  
3670  
3675  
3680  
3685  
3690  
3695  
3700  
3705  
3710  
3715  
3720  
3725  
3730  
3735  
3740  
3745  
3750  
3755  
3760  
3765  
3770  
3775  
3780  
3785  
3790  
3795  
3800  
3805  
3810  
3815  
3820  
3825  
3830  
3835  
3840  
3845  
3850  
3855  
3860  
3865  
3870  
3875  
3880  
3885  
3890  
3895  
3900  
3905  
3910  
3915  
3920  
3925  
3930  
3935  
3940  
3945  
3950  
3955  
3960  
3965  
3970  
3975  
3980  
3985  
3990  
3995  
4000  
4005  
4010  
4015  
4020  
4025  
4030  
4035  
4040  
4045  
4050  
4055  
4060  
4065  
4070  
4075  
4080  
4085  
4090  
4095  
4100  
4105  
4110  
4115  
4120  
4125  
4130  
4135  
4140  
4145  
4150  
4155  
4160  
4165  
4170  
4175  
4180  
4185  
4190  
4195  
4200  
4205  
4210  
4215  
4220  
4225  
4230  
4235  
4240  
4245  
4250  
4255  
4260  
4265  
4270  
4275  
4280  
4285  
4290  
4295  
4300  
4305  
4310  
4315  
4320  
4325  
4330  
4335  
4340  
4345  
4350  
4355  
4360  
4365  
4370  
4375  
4380  
4385  
4390  
4395  
4400  
4405  
4410  
4415  
4420  
4425  
4430  
4435  
4440  
4445  
4450  
4455  
4460  
4465  
4470  
4475  
4480  
4485  
4490  
4495  
4500  
4505  
4510  
4515  
4520  
4525  
4530  
4535  
4540  
4545  
4550  
4555  
4560  
4565  
4570  
4575  
4580  
4585  
4590  
4595  
4600  
4605  
4610  
4615  
4620  
4625  
4630  
4635  
4640  
4645  
4650  
4655  
4660  
4665  
4670  
4675  
4680  
4685  
4690  
4695  
4700  
4705  
4710  
4715  
4720  
4725  
4730  
4735  
4740  
4745  
4750  
4755  
4760  
4765  
4770  
4775  
4780  
4785  
4790  
4795  
4800  
4805  
4810  
4815  
4820  
4825  
4830  
4835  
4840  
4845  
4850  
4855  
4860  
4865  
4870  
4875  
4880  
4885  
4890  
4895  
4900  
4905  
4910  
4915  
4920  
4925  
4930  
4935  
4940  
4945  
4950  
4955  
4960  
4965  
4970  
4975  
4980  
4985  
4990  
4995  
5000  
5005  
5010  
5015  
5020  
5025  
5030  
5035  
5040  
5045  
5050  
5055  
5060  
5065  
5070  
5075  
5080  
5085  
5090  
5095  
5100  
5105  
5110  
5115  
5120  
5125  
5130  
5135  
5140  
5145  
5150  
5155  
5160  
5165  
5170  
5175  
5180  
5185  
5190  
5195  
5200  
5205  
5210  
5215  
5220  
5225  
5230  
5235  
5240  
5245  
5250  
5255  
5260  
5265  
5270  
5275  
5280  
5285  
5290  
5295  
5300  
5305  
5310  
5315  
5320  
5325  
5330  
5335  
5340  
5345  
5350  
5355  
5360  
5365  
5370  
5375  
5380  
5385  
5390  
5395  
5400  
5405  
5410  
5415  
5420  
5425  
5430  
5435  
5440  
5445  
5450  
5455  
5460  
5465  
5470  
5475  
5480  
5485  
5490  
5495  
5500  
5505  
5510  
5515  
5520  
5525  
5530  
5535  
5540  
5545  
5550  
5555  
5560  
5565  
5570  
5575  
5580  
5585  
5590  
5595  
5600  
5605  
5610  
5615  
5620  
5625  
5630  
5635  
5640  
5645  
5650  
5655  
5660  
5665  
5670  
5675  
5680  
5685  
5690  
5695  
5700  
5705  
5710  
5715  
5720  
5725  
5730  
5735  
5740  
5745  
5750  
5755  
5760  
5765  
5770  
5775  
5780  
5785  
5790  
5795  
5800  
5805  
5810  
5815  
5820  
5825  
5830  
5835  
5840  
5845  
5850  
5855  
5860  
5865  
5870  
5875  
5880  
5885  
5890  
5895  
5900  
5905  
5910  
5915  
5920  
5925  
5930  
5935  
5940  
5945  
5950  
5955  
5960  
5965  
5970  
5975  
5980  
5985  
5990  
5995  
6000  
6005  
6010  
6015  
6020  
6025  
6030  
6035  
6040  
6045  
6050  
6055  
6060  
6065  
6070  
6075  
6080  
6085  
6090  
6095  
6100  
6105  
6110  
6115  
6120  
6125  
6130  
6135  
6140  
6145  
6150  
6155  
6160  
6165  
6170  
6175  
6180  
6185  
6190  
6195  
6200  
6205  
6210  
6215  
6220  
6225  
6230  
6235  
6240  
6245  
6250  
6255  
6260  
6265  
6270  
6275  
6280  
6285  
6290  
6295  
6300  
6305  
6310  
6315  
6320  
6325  
6330  
6335  
6340  
6345  
6350  
6355  
6360  
6365  
6370  
6375  
6380  
6385  
6390  
6395  
6400  
6405  
6410  
6415  
6420  
6425  
6430  
6435  
6440  
6445  
6450  
6455  
6460  
6465  
6470  
6475  
6480  
6485  
6490  
6495  
6500  
6505  
6510  
6515  
6520  
6525  
6530  
6535  
6540  
6545  
6550  
6555  
6560  
6565  
6570  
6575  
6580  
6585  
6590  
6595  
6600  
6605  
6610  
6615  
6620  
6625  
6630  
6635  
6640  
6645  
6650  
6655  
6660  
6665  
6670  
6675  
6680  
6685  
6690  
6695  
6700  
6705  
6710  
6715  
6720  
6725  
6730  
6735  
6740  
6745  
6750  
6755  
6760  
6765  
6770  
6775  
6780  
6785  
6790  
6795  
6800  
6805  
6810  
6815  
6820  
6825  
6830  
6835  
6840  
6845  
6850  
6855  
6860  
6865  
6870  
6875  
6880  
6885  
6890  
6895  
6900  
6905  
6910  
6915  
6920  
6925  
6930  
6935  
6940  
6945  
6950  
6955  
6960  
6965  
6970  
6975  
6980  
6985  
6990  
6995  
7000  
7005  
7010  
7015  
7020  
7025  
7030  
7035  
7040  
7045  
7050  
7055  
7060  
7065  
7070  
7075  
7080  
7085  
7090  
7095  
7100  
7105  
7110  
7115  
7120  
7125  
7130  
7135  
7140  
7145  
7150  
7155  
7160  
7165  
7170  
7175  
7180  
7185  
7190  
7195  
7200  
7205  
7210  
7215  
7220  
7225  
7230  
7235  
7240  
7245  
7250  
7255  
7260  
7265  
7270  
7275  
7280  
7285  
7290  
7295  
7300  
7305  
7310  
7315  
7320  
7325  
7330  
7335  
7340  
7345  
7350  
7355  
7360  
7365  
7370  
7375  
7380  
7385  
7390  
7395  
7400  
7405  
7410  
7415  
7420  
7425  
7430  
7435  
7440  
7445  
7450  
7455  
7460  
7465  
7470  
7475  
7480  
7485  
7490  
7495  
7500  
7505  
7510  
7515  
7520  
7525  
7530  
7535  
7540  
7545  
7550  
7555  
7560  
7565  
7570  
7575  
7580  
7585  
7590  
7595  
7600  
7605  
7610  
7615  
7620  
7625  
7630  
7635  
7640  
7645  
7650  
7655  
7660  
7665  
7670  
7675  
7680  
7685  
7690  
7695  
7700  
7705  
7710  
7715  
7720  
7725  
7730  
7735  
7740  
7745  
7750  
7755  
7760  
7765  
7770  
7775  
7780  
7785  
7790  
7795  
7800  
7805  
7810  
7815  
7820  
7825  
7830  
7835  
7840  
7845  
7850  
7855  
7860  
7865  
7870  
7875  
7880  
7885  
7890  
7895  
7900  
7905  
7910  
7915  
7920  
7925  
7930  
7935  
7940  
7945  
7950  
7955  
7960  
7965  
7970  
7975  
7980  
7985  
7990  
7995  
8000  
8005  
8010  
8015  
8020  
8025  
8030  
8035  
8040  
8045  
8050  
8055  
8060  
8065  
8070  
8075  
8080  
8085  
8090  
8095  
8100  
8105  
8110  
8115  
8120  
8125  
8130  
8135  
8140  
8145  
8150  
8155  
8160  
8165  
8170  
8175  
8180  
8185  
8190  
8195  
8200  
8205  
8210  
8215  
8220  
8225  
8230  
8235  
8240  
8245  
8250  
8255  
8260  
8265  
8270  
8275  
8280  
8285  
8290  
8295  
8300  
8305  
8310  
8315  
8320  
8325  
8330  
8335  
8340  
8345  
8350  
8355  
8360  
8365  
8370  
8375  
8380  
8385  
8390  
8395  
8400  
8405  
8410  
8415  
8420  
8425  
8430  
8435  
8440  
8445  
8450  
8455  
8460  
8465  
8470  
8475  
8480  
8485  
8490  
8495  
8500  
8505  
8510  
8515  
8520  
8525  
8530  
8535  
8540  
8545  
8550  
8555  
8560  
8565  
8570  
8575  
8580  
8585  
8590  
8595  
8600  
8605  
8610  
8615  
8620  
8625  
8630  
8635  
8640  
8645  
8650  
8655  
8660  
8665  
8670  
8675  
8680  
8685  
8690  
8695  
8700  
8705  
8710  
8715  
8720  
8725  
8730  
8735  
8740  
8745  
8750  
8755  
8760  
8765  
8770  
8775  
8780  
8785  
8790  
8795  
8800  
8805  
8810  
8815  
8820  
8825  
8830  
8835  
8840  
8845  
8850  
8855  
8860  
8865  
8870  
8875  
8880  
8885  
8890  
8895  
8900  
8905  
8910  
8915  
8920  
8925  
8930  
8935  
8940  
8945  
8950  
8955  
8960  
8965  
8970  
8975  
8980  
8985  
8990  
8995  
9000  
9005  
9010  
9015  
9020  
9025  
9030  
9035  
9040  
9045  
9050  
9055  
9060  
9065  
9070  
9075  
9080  
9085  
9090  
9095  
9100  
9105  
9110  
9115  
9120  
9125  
9130  
9135  
9140  
9145  
9150  
9155  
9160  
9165  
9170  
9175  
9180  
9185  
9190  
9195  
9200  
9205  
9210  
9215  
9220  
9225  
9230  
9235  
9240  
9245  
9250  
9255  
9260  
9265  
9270  
9275  
9280  
9285  
9290  
9295  
9300  
9305  
9310  
9315  
9320  
9325  
9330  
9335  
9340  
9345  
9350  
9355  
9360  
9365  
9370  
9375  
9380  
9385  
9390  
9395  
9400  
9405  
9410  
9415  
9420  
9425  
9430  
9435  
9440  
9445  
9450  
9455  
9460  
9465  
9470  
9475  
9480  
9485  
9490  
9495  
9500  
9505  
9510  
9515  
9520  
9525  
9530  
9535  
9540  
9545  
9550  
9555  
9560  
95

FIG. 17 is a block diagram illustrating an encryption system according to one embodiment of the invention. As described above, a separate inbound controller may be provided for each inbound port and a separate outbound controller may be provided for each outbound port. As shown, an inbound controller 1702 is provided for an inbound port. As packets are received by the inbound controller 1702, they may be classified and stored in the appropriate inbound queue by inbound classifier 1704. CPU 1706 may then transfer entire inbound queues stored in memory 1708 to outbound controller 1710 which is coupled to encryption box 1712. Thus, the encryption box 1712 may encrypt each queue rather than separate packets. The encryption box 1712 may then store each encrypted queue in the corresponding inbound queue or compatible data structure. Since the packets have been classified in appropriate inbound queues, encryption keys may therefore be distinguished. Since encryption may vary according to source address, destination address, and other factors, the outbound controller 1712 may classify each inbound queue in multiple outbound queues. In this manner, each receiving service may ascertain the status of encryption of each queue. The outbound controller 1712 may classify each inbound queue in the appropriate outbound queue through the use of a separate outbound classifier 1714. Thus, a separate classifier may be provided for each controller or port. Alternatively, the outbound queue classification may be made implicit through storing an outbound queue ID in the inbound queue header by the inbound classifier 1704. Accordingly, since encryption is performed for each queue rather than on a per packet basis, CPU overhead is reduced. In addition to being implemented in encryption systems, the present invention may be used in various systems such as data compression and decompression systems.

The present invention reduces CPU overhead in the forwarding process. This is accomplished through transferring sets of packets, or queues of packets, between an inbound controller and an outbound controller. Thus, one transfer operation may be performed for a set of packets rather than a set of operations for one packet. Moreover, operations utilized to de-allocate packets may be

similarly reduced. Accordingly, CPU overhead is reduced, increasing the packet per second forwarding rate.

In addition to reducing CPU overhead, memory utilization is minimized through the allocation of packet buffers by the inbound controller. Since the packet buffers are de-allocated upon transmission, memory is efficiently utilized. As a result, the amount of memory required and production costs associated with each router are minimized.

Through combining software and hardware modules, the present invention provides efficient and accurate packet forwarding. Hardware is efficient, but typically does not provide global knowledge. Thus hardware may be utilized to provide speed where global knowledge is not required. By way of example, a hardware classifier may provide an increased forwarding rate. At the same time, software provides global knowledge and intelligence in forwarding decisions to increase the efficiency of the forwarding process. By way of example, where a particular outbound port is congested, a queue may be discarded by the CPU.

The invention can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data which can thereafter be read by a computer system. Examples of the computer readable medium include read-only memory, random-access memory, CD-ROMs, magnetic tape, and optical data storage devices.

Although illustrative embodiments and applications of this invention are shown and described herein, many variations and modifications are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those of ordinary skill in the art after perusal of this application. For instance, the present invention is described as utilizing a single outbound queue for each outbound port. However, it should be understood that the present invention is not limited to this arrangement, but

instead would equally apply regardless of the number of outbound queues provided per port. Also, the invention is described as providing a single inbound controller for each inbound port. Similarly, the invention is described as providing a single outbound controller for each outbound port. This is not a requirement of the present invention and therefore, other arrangements for the inbound and outbound controllers would still fall within the scope of the invention. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

5

10

600600-00275700